

Estimated VC dimension for risk bounds

Daniel J. McDonald Cosma Rohilla Shalizi
 Carnegie Mellon University Carnegie Mellon University
danielmc@cmu.edu cshalizi@cmu.edu

Mark Schervish
 Carnegie Mellon University
mark@cmu.edu

Version: November 16, 2011

Abstract

Vapnik-Chervonenkis (VC) dimension is a fundamental measure of the generalization capacity of learning algorithms. However, apart from a few special cases, it is hard or impossible to calculate analytically. Vapnik et al. [10] proposed a technique for estimating the VC dimension empirically. While their approach behaves well in simulations, it could not be used to bound the generalization risk of classifiers, because there were no bounds for the estimation error of the VC dimension itself. We rectify this omission, providing high probability concentration results for the proposed estimator and deriving corresponding generalization bounds.

1 Introduction

Statistical learning theory is fundamentally concerned with picking, out of some class of plausible or convenient models, ones whose predictions will be nearly optimal. Statistical optimality is most often demonstrated by controlling the risk, or generalization error, of predictive models, i.e., their expected inaccuracy on new data from the same source as that used to fit the model. The paradigmatic case confronts the learner with a labeled set of training examples $Z = \{(y_1, x_1), \dots, (y_n, x_n)\}$ drawn independently from a distribution μ over $\mathcal{Y} \times \mathcal{X}$. For concreteness, we take the standard task of pattern recognition with vector features, setting $\mathcal{Y} = \{0, 1\}$ and $\mathcal{X} = \mathbb{R}^p$. Our contribution is to controlling the risk of pattern recognition when using analytically intractable models.

Consider a class \mathcal{F} of possible predictors, that is a collection of functions from \mathcal{X} to \mathcal{Y} . From this class, the learner uses the training set to choose some $f \in \mathcal{F}$, hoping to make as few errors in the future as possible when facing similar data. This amounts to controlling the *risk* of f

$$R_n(f) = \mathbb{E}_\mu[I(Y \neq f(X))], \quad (1)$$

where $I(A)$ is the indicator of the event A . Since the distribution μ is unknown, the risk cannot be calculated explicitly, so learners often proxy it by the *empirical risk* of f ,

$$\hat{R}_n(f, Z) = \frac{1}{n} \sum_{i=1}^n I(Y_i \neq f(X_i)), \quad (2)$$

which we will abbreviate $\widehat{R}_n(f)$ when possible. Since (2) approximates (1), we can choose a good predictor \widehat{f} by solving

$$\widehat{f} = \operatorname{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n I(Y_i \neq f(X_i)).$$

This process is *empirical risk minimization*, or ERM. ERM itself is quite general, and with appropriate loss functions includes ordinary least squares regression, maximum likelihood, nonparametric density estimation, and M -estimation.

The next step in the statistical learning paradigm is to evaluate the performance of ERM. Is \widehat{f} consistent (in risk) for f ? What is the rate of convergence? Can we control the generalization error of the chosen \widehat{f} ? In fact, all of these questions are answered. Vapnik and Chervonenkis [9] gave necessary and sufficient conditions for uniform convergence of $\widehat{R}_n(f)$ to $R_n(f)$ in terms of the VC entropy. However, the VC entropy itself depends on the unknown distribution μ . To get around this, we look instead at a bound for the VC entropy which is uniform over probability measures: the *growth function*, which can be calculated from the VC dimension, which is based on the shattering coefficient.

Definition 1.1. Let \mathbb{U} be some (infinite) set and let \mathcal{S} be a finite subset of \mathbb{U} . Let \mathcal{C} be a family of subsets of \mathbb{U} . We say that \mathcal{C} shatters \mathcal{S} if for every $\mathcal{S}' \subseteq \mathcal{S}$, $\exists C \in \mathcal{C}$ such that $\mathcal{S}' = \mathcal{S} \cap C$.

Definition 1.2 (VC dimension). The Vapnik-Chervonenkis (VC) dimension of \mathcal{C} is

$$\text{VCD}(\mathcal{C}) := \sup \{\text{card } \mathcal{S} : \mathcal{S} \text{ is shattered by } \mathcal{C}\}.$$

Application of VC dimension to classes of functions is reasonably straightforward for pattern recognition. To $f \in \mathcal{F}$, associate the set $C_f = \{u \in \mathbb{U} : f(u) = 1\}$, and associate to \mathcal{F} the class $\mathcal{C}_{\mathcal{F}} := \{C_f : f \in \mathcal{F}\}$. Then define $\text{VCD}(\mathcal{F}) := \text{VCD}(\mathcal{C}_{\mathcal{F}})$.

VC dimension is just one of many ways to measure the richness or complexity of a class of functions. Others include covering numbers, Pseudo-dimension [3], fat-shattering dimension, and Rademacher complexity [2]. Heuristically, larger complexity leads to smaller minimum risk but higher estimation variance, and thus it is important to balance the complexity of the function class with the amount of data available. For VC dimension, Vapnik [8] shows that a sufficient condition for uniform risk consistency is that

$$\lim_{n \rightarrow \infty} \frac{\log GF(h^*, n)}{n} = 0,$$

where $\log GF(h, n) \leq h(\log(n/h) + 1)$ is the growth function and $h^* = \text{VCD}(\mathcal{F})$ is the VC dimension of the function class. Furthermore, Vapnik [7, 8] proves a concentration result of the empirical risk around the true risk: for any $\rho > 0$

$$\mathbb{P}_{\mu} \left(\sup_{f \in \mathcal{F}} |R_n(f) - \widehat{R}_n(f)| > \rho \right) < 4GF(h^*, 2n) \exp \{-n\rho^2\}. \quad (3)$$

Similar bounds exist for other loss functions such as margin loss, loss functions constrained to a compact interval, or extended real-valued loss functions for regression problems.

Given a function class \mathcal{F} , knowing $h^* = \text{VCD}(\mathcal{F})$ is crucial to using these sorts of results. However, for many interesting function classes (support vector machines, multi-layer neural networks, random forests, etc.) this knowledge is entirely unavailable. The combinatorial nature of VC dimension makes it very difficult to find in interesting cases. As a remedy, Vapnik et al. [10]

propose a way to estimate the VC dimension by simulation. While the authors showed its accuracy by estimating the VC dimension of linear classifiers (known to be the number of covariates with an extra degree of freedom for the intercept), estimated VC dimension cannot be simply plugged in to finite-sample concentration results (such as (3)), because the estimates themselves fluctuate around the true values. Since VC dimension is only useful to the extent it lets us bound generalization risk, this presents a problem. In this paper, we rectify this situation.

We prove two main results. First, we show that, using the procedure of [10], the estimated VC dimension, \hat{h} , will concentrate around the truth, h^* , with high probability:

Theorem 1.3. *Let $\delta > \frac{4}{\sqrt{2mk}} \max\{24c_1, 29\}$ and suppose that $h^* \leq M$. Then*

$$\mathbb{P}\left(|\hat{h} - h^*| > \delta\right) \leq 13 \exp\left\{-\frac{mkc_2\delta^2}{16c_3}\right\}$$

where c_1 , c_2 , and c_3 are constants given in the proof and in Table 1, and k and m are integers freely chosen as part of the simulation procedure.

Second, we show that if we use the estimated VC dimension, we can still recover bounds like that in (3):

Theorem 1.4. *Choose δ as in Theorem 1.3. Let $\rho > 0$. Set*

$$\varphi = 13 \exp\left\{-\frac{mkc_2\delta^2}{16c_3}\right\}.$$

Then, for any classifier $f \in \mathcal{F}$ where \mathcal{F} has estimated VC dimension \hat{h} , we have

$$\mathbb{P}\left(\sup_{f \in \mathcal{F}} |R_n(f) - \hat{R}_n(f)| > \rho\right) \leq 4GF(\hat{h} + \delta, 2n) \exp\{-n\rho^2\}(1 - \varphi) + \varphi. \quad (4)$$

The first term on the right of (4) is the same as the original bound in (3), except that the true VC dimension is replaced with its estimate \hat{h} plus a small fudge factor δ . The second term depends on the confidence that we have in our estimate, through φ . The estimation procedure allows us to estimate h^* arbitrarily well, given infinite computational time, through the choice of m and k . Of course this is infeasible in practice, but Theorem 1.4 allows the user to trade computational time for statistical accuracy.

The remainder of this paper provides details for the proofs of our two main theorems. Section 2 summarizes the estimation procedure developed in Vapnik et al. [10]. Section 3 proves both theorems, drawing on empirical process theory. Because there is a lot of notation, we summarize it in Table 1. Finally, Section 4 concludes and provides some ideas for future work.

2 Estimation

Vapnik et al. [10] show that the expected maximum deviation between the empirical risks of a classifier on two datasets can be bounded by a function which depends only on the VC dimension of the classifier. In other words, given a collection of classifiers \mathcal{F} , and two data sets $W = \{(y_1, x_1), \dots, (y_n, x_n)\}$ and $W' = \{(y'_1, x'_1), \dots, (y'_n, x'_n)\}$, we have the bound

$$\xi(n) := \mathbb{E}\left[\sup_{f \in \mathcal{F}} (\hat{R}_n(f, W) - \hat{R}_n(f, W'))\right] \leq \begin{cases} 1 & n/h^* \leq \frac{1}{2} \\ C_1 \frac{\log(2n/h^*)+1}{n/h^*} & \text{if } n/h^* \text{ is small} \\ C_2 \sqrt{\frac{\log(2n/h^*)+1}{n/h^*}} & \text{if } n/h^* \text{ is large.} \end{cases} \quad (5)$$

Table 1: Constants and important notation

Notation	Meaning
h^*	the VC dimension of the function class \mathcal{F}
\hat{h}	the estimate of VC dimension via (3)
M	we assume $h^* \leq M$
$\Phi_h(n)$	$\begin{cases} 1 & n < h/2 \\ a^{\frac{\log \frac{2n}{h} + 1}{\frac{n}{h} - a''}} \left(\sqrt{1 + \frac{a'(\frac{n}{h} - a'')}{\log \frac{2n}{h} + 1}} + 1 \right) & \text{else.} \end{cases}$
a	0.16
a'	1.2
a''	0.14927
φ	$13 \exp \left\{ -\frac{mkc_2\delta^2}{16c_3} \right\}$
$GF(h, n)$	$\leq h(\log(n/h) + 1)$
$c(n, M)$	$\left\{ \begin{array}{l} \text{Lipschitz-like constants such that } \forall n: \\ c(n, M) h - h' \leq \Phi_h(n) - \Phi_{h'}(n) \leq L(n) h - h' \end{array} \right.$
$L(n)$	
$N(\eta, \mathcal{G})$	the η -covering number of \mathcal{G}
$H(\eta, \mathcal{G})$	the η -entropy of \mathcal{G}
k, m	integers chosen for the simulation in Algorithm 1
c_1	$(c' + 1/4)\sqrt{\log(4c' + 1)} - \frac{\sqrt{\pi}}{8}\text{erfi}(\sqrt{4c' + 1})$
c'	$\frac{1}{k} \sum_{\ell=1}^k L^2(n_\ell)$
c_2	$\frac{1}{k} \sum_{\ell=1}^k c^2(n_\ell, M)$
c_3	2304

We can bound (5) by $\Phi_{h^*}(n)$, viewed as a function of n and parametrized by h :

$$\Phi_h(n) = \begin{cases} 1 & n < h/2 \\ a^{\frac{\log \frac{2n}{h} + 1}{\frac{n}{h} - a''}} \left(\sqrt{1 + \frac{a'(\frac{n}{h} - a'')}{\log \frac{2n}{h} + 1}} + 1 \right) & \text{else.} \end{cases} \quad (6)$$

Here the constants $a = 0.16$, $a' = 1.2$ were determined numerically in [10] to adjust the trade-off between “small” and “large” in (5), and $a'' = 0.14927$ was chosen so that $\Phi(0.5) = 1$ (this choice depends only on a and a''). Furthermore, the bound is tight. Since (6) is known up to h , we can estimate it given knowledge of the maximum deviation on the left side of (5). Of course, we do not

Algorithm 1 Generate $\widehat{\xi}(n_\ell)$

Given a collection of possible classifiers \mathcal{F} and a grid of design points n_1, \dots, n_k , generate $\widehat{\xi}(n_\ell)$. Repeat the procedure at each design point, n_ℓ , m times.

- 1: Generate a data set from the same sample space $\mathcal{Y} \times \mathcal{X}$ as the training sample that is independent of the training sample. The generated set should be of size $2n_\ell$: $\{(y_1, x_1), \dots, (y_{2n_\ell}, x_{2n_\ell})\}$.
 - 2: Split the data set into two equal sets, W and W' .
 - 3: Flip the labels (y values) of W' .
 - 4: Merge the two sets and train the classifier simultaneously on the entire set: W with the “correct” labels and W' with the “wrong” labels.
 - 5: Calculate the training error of the estimated classifier \widehat{f} on W with the ‘correct’ labels and on W' using the “correct” labels.
 - 6: Set $\widehat{\xi}_i(n_\ell) = |\widehat{R}_{n_\ell}(\widehat{f}, W) - \widehat{R}_{n_\ell}(\widehat{f}, W')|$.
 - 7: Set $\widehat{\xi}(n_\ell) = \frac{1}{m} \sum_{i=1}^m \widehat{\xi}_i(n_\ell)$.
-

have such knowledge, but we can generate observations

$$\widehat{\xi}(n) = \Phi_h(n) + \epsilon(n)$$

at design points n . Here ϵ is mean zero noise (since the bound is tight) having an unknown distribution with support on $[0, 1]$. Given enough such observations at different design points n_ℓ , we can then estimate the true VC dimension h^* using nonlinear least squares. Of course, generating $\widehat{\xi}(n_\ell)$ is nontrivial. Vapnik et al. [10] give an algorithm for generating the appropriate observations. Essentially, at each (fixed) design point $n_\ell : \ell \in \{1, \dots, k\}$, we simulate m data points $(\widehat{\xi}_i(n_\ell), \Phi_h(n_\ell))$, for $i = 1, \dots, m$, so as to approximate $\xi(n_\ell)$ as defined in (5). This procedure is shown in Algorithm 1. Vapnik et al. [10] show that this algorithm works well in practice, recovering the known VC dimension of linear classifiers ($p + 1$ for p explanatory variables and an intercept) and demonstrating that the method for generating the dataset does not affect the algorithm’s performance.¹ In the next section, we prove our main result, showing that in fact, the estimate concentrates around the truth with high probability.

3 Proof of results

We now prove Theorem 1.3 and Theorem 1.4. The proofs draw heavily on the empirical process techniques of van de Geer [5] and van de Geer [6]; however, those works ignored constants, and made stronger assumptions than necessary for the case at hand. We strive to make our results as self-contained as possible, appealing to [6] only for the proof of Corollary 3.5.

Our goal is to show that the estimated VC dimension \widehat{h} is close to the true dimension h^* . This will mean showing that $\Phi_{\widehat{h}}$ is close to Φ_{h^*} when averaged over the design points n_ℓ . It will be

¹There are of course ways to generate data in so that this procedure will fail, e.g., generating the data with too-regular determinism, or with dependence. We refer the cautious reader to Vapnik et al. [10]. We also return to this point at the end of §3.

convenient to introduce a norm and inner product for functions $g : \mathbb{R} \rightarrow \mathbb{R}$:

$$\begin{aligned} \|g\|_k^2 &= \frac{1}{k} \sum_{\ell=1}^k g(n_\ell)^2 \\ (\epsilon, g)_k &= \frac{1}{k} \sum_{\ell=1}^k \epsilon(n_\ell) g(n_\ell). \end{aligned}$$

So we take as our estimate of h^*

$$\hat{h} = \operatorname{argmin}_{h \in [0, M]} \left\| \hat{\xi} - \Phi_h \right\|_k,$$

and our immediate goal is control over $\left\| \Phi_{\hat{h}} - \Phi_{h^*} \right\|_k^2$.

For every $f \in \mathcal{F}$ and every dataset Z , $\hat{R}_n(f, W)$ is bounded between 0 and 1. Therefore, the residuals $\epsilon(n_\ell)$ are also in $[0, 1]$. In fact, we can show that they are subgaussian.

Lemma 3.1. *At all design points n ,*

$$\mathbb{E}[\exp\{t\epsilon(n)\}] \leq \exp\{t^2/8m\}. \quad (7)$$

Proof. By a standard Hoeffding type argument, we have that

$$\mathbb{E}[\exp\{t\epsilon_i(n)\}] \leq \exp\{t^2/8\}.$$

Therefore

$$\begin{aligned} \mathbb{E}[\exp\{t\epsilon(n)\}] &= \mathbb{E} \left[\prod_{i=1}^m \exp\{t\epsilon_i(n)/n\} \right] \\ &\leq \exp \left\{ \frac{mt^2}{m^2 8} \right\} \\ &= \exp\{t^2/8m\}. \end{aligned}$$

□

The next step is to show that we can control weighted averages of the $\epsilon(n_\ell)$.

Lemma 3.2. *Suppose $\epsilon_\ell := \epsilon(n_\ell)$ are random variables satisfying (7). Then for any $\gamma \in \mathbb{R}^k$ and $\rho > 0$,*

$$\mathbb{P} \left(\left| \sum_{\ell=1}^k \epsilon_\ell \gamma_\ell \right| > \rho \right) \leq 2 \exp \left\{ - \frac{2m\rho^2}{\sum_{\ell=1}^k \gamma_\ell^2} \right\}.$$

Proof. Using a Chernoff bound, we have, for $t > 0$

$$\mathbb{P} \left(\sum_{\ell=1}^k \epsilon_\ell \gamma_\ell > \rho \right) \leq \exp \left\{ -t\rho + \sum_{\ell=1}^k \frac{\gamma_\ell t^2}{8m} \right\}.$$

Taking

$$t = \frac{4m\rho}{\sum_{\ell=1}^k \gamma_\ell}$$

minimizes the right hand side. The same argument applies for $-\sum_{\ell=1}^k \epsilon_\ell \gamma_\ell$, so a union bound gives the result. □

In order to state our result about $\|\Phi_{\hat{h}} - \Phi_{h^*}\|_k$, we must specify the complexity of the function class $\mathcal{G} := \{\Phi_h : 0 \leq h \leq M\}$, which we will measure with its *entropy*.

Definition 3.3. *The functions g_1, \dots, g_n are an η -cover of \mathcal{G} if every $g \in \mathcal{G}$ is within η of some g_j , $\|g - g_j\|_k \leq \eta$. The η -covering number $N(\eta, \mathcal{G})$ is the cardinality of the smallest η -cover (or ∞ if there isn't one). The η -entropy is the log of the covering number, $H(\eta, \mathcal{G}) = \log N(\eta, \mathcal{G})$.*

While it may seem excessive to use covering numbers and entropy to deal with a function class parametrized by a scalar, doing so lets us get much tighter bounds than would otherwise be possible. The key to our argument will be the entropy of the restricted class $\mathcal{G}(\tau) := \{\Phi_h \in \mathcal{G} : \|\Phi - \Phi_{h^*}\|_k \leq \tau\}$.

Lemma 3.4.

$$H(\eta, \mathcal{G}(\tau)) \leq \log \left(\frac{4\tau/c' + \eta}{\eta} \right),$$

where c' is defined below.

Proof. Φ_h is bounded and differentiable in h and therefore Lipschitz with constants $L(n)$. Thus

$$\|\Phi_h - \Phi_{h'}\|_k \leq \frac{1}{k} \sum_{\ell=1}^k L^2(n_\ell) |h - h'|.$$

Set $c' = \frac{1}{k} \sum_{\ell=1}^k L^2(n_\ell)$. Covering a τ ball around Φ_h in the $\|\cdot\|_k$ metric is then equivalent to covering a τ/c' ball around h in the Euclidean metric. It is well known (cf. [6]) that

$$H(\eta, B(\tau/c')) \leq \log \left(\frac{4\tau/c' + \eta}{\eta} \right).$$

□

The remaining proofs rely on the *peeling device*. Intuitively, the idea is that considering the entropy of larger and larger balls centered around Φ_{h^*} will allow us to “peel” off sets of increasingly smaller probability. This peeling argument is critical to our proof that $\|\Phi_{\hat{h}} - \Phi_{h^*}\|_k$ is small with high probability.

To use peeling here, define $d(h) := \|\Phi_h - \Phi_{h^*}\|_k$ and consider a strictly increasing sequence v_s , starting with $v_0 = 0$ but growing to ∞ . We can peel \mathcal{G} into $\mathcal{G} = \bigcup_{s=1}^{\infty} \mathcal{G}_s$, where

$$\mathcal{G}_s = \{\Phi_h \in \mathcal{G} : v_{s-1} \leq d(h) < v_s\}.$$

Then we have that for any $\rho > 0$, and our residuals ϵ (which implicitly depend on the choice of $g := \Phi_h$),

$$\mathbb{P} \left(\sup_{g \in \mathcal{G}} \frac{|\epsilon|}{d(h)} > \rho \right) \leq \sum_{s=1}^{\infty} \mathbb{P} \left(\sup_{g \in \mathcal{G}_s} \frac{|\epsilon|}{d(h)} > \rho \right) \leq \sum_{s=1}^{\infty} \mathbb{P} \left(\sup_{g \in \mathcal{G}, d(h) < v_s} |\epsilon| > \rho v_{s-1} \right).$$

This lets us get probability inequalities for the weighted process from probabilities for the original process. We will want to allow the weights γ_ℓ in Lemma 3.2 to depend on functions Φ_h , and in particular investigate the behavior of the worst-case h . Taking $v_s = 2^s$ for $s > 0$ and $v_0 = 0$ will allow us to derive an important corollary to Lemma 3.2 as well as Theorem 3.6.

Choosing v_s this way means that it is not enough to control the covering number of the entire function class \mathcal{G} , but rather we must cover a sequence of restricted classes $\mathcal{G}(\tau)$ with smaller and smaller balls. Therefore, we will need the entropy sum,

$$J(\tau) := \sum_{s=1}^{\infty} 2^{-s} \tau \sqrt{H(2^{-s} \tau, \mathcal{G}(\tau))}.$$

which is bounded by the entropy integral,

$$J(\tau) \leq 2 \int_0^{\tau} du \sqrt{H(u, \mathcal{G}(\tau))}$$

(see [6, p. 29]). Lemma 3.4 implies that²

$$J(\tau) \leq 2\tau \int_0^1 dv \sqrt{\log(1 + 4v/c')} \leq 2c_1 \tau.$$

Finally, we can prove an important corollary to Lemma 3.2. The proof makes use of the entropy integral as well as the peeling device, and it follows from Lemma 3.2 in van de Geer [6], so we provide only the necessary adjustments in our proof here. However, we will need both the peeling device and the entropy integral again in the proof of Theorem 3.6.

Corollary 3.5 (Corollary of Lemma 3.2). *If $\sup_{g \in \mathcal{G}} \|g\|_k \leq \tau$ and (7) holds for all design n_ℓ , then for all*

$$\delta > \frac{\tau}{\sqrt{2km}} \max\{24c_1, 29\},$$

we have, as a consequence of Lemma 3.2,

$$\mathbb{P} \left(\sup_{g \in \mathcal{G}} \left| \frac{1}{k} \sum_{\ell=1}^k \epsilon_\ell g(n_\ell) \right| > \delta \right) \leq 4 \exp \left\{ -\frac{km\delta^2}{c_3 \tau^2} \right\},$$

where c_1 is as above and $c_3 = 2304$.

Proof. The proof is given in Lemma 3.2 in van de Geer [6]. In our case, the entropy integral converges, so we may take $K = \infty$ in that proof. Furthermore, we replace equation (3.3) there with the result of Lemma 3.2 here, and set $\epsilon = \delta/2$. \square

Theorem 3.6. *Suppose that \hat{h} is the solution to (3). Let*

$$\delta > \frac{4}{\sqrt{2mk}} \max\{24c_1, 29\}. \tag{8}$$

Then,

$$\mathbb{P}(\|\Phi_{\hat{h}} - \Phi_{h^*}\|_k > \delta) \leq 13 \exp \left\{ -\frac{mk\delta^2}{16c_3} \right\}.$$

²In fact, $c_1 = (c' + 1/4)\sqrt{\log(4c' + 1)} - \frac{\sqrt{\pi}}{8} \operatorname{erfi}(\sqrt{4c' + 1})$, where erfi is the imaginary error function. Despite the adjective “imaginary”, c_1 is always real.

Proof. First, note that $\|\Phi_{\hat{h}} - \Phi_{h^*}\|_k^2 \leq 2(w, \Phi_{\hat{h}} - \Phi_{h^*})_k$. Then we use peeling and the lemmas above:

$$\begin{aligned} & \mathbb{P}(\|\Phi_{\hat{h}} - \Phi_{h^*}\|_k > \delta) \\ & \leq \sum_{s=0}^{\infty} \mathbb{P}\left(\sup_{\Phi_h \in \mathcal{G}(2^{s+1}\delta)} (w, \Phi_h - \Phi_{h^*})_k > 2^{2s-1}\delta^2\right) \\ & = \sum_{s=0}^{\infty} \mathbb{P}_s. \end{aligned}$$

Now $\forall s \geq 0$,

$$2^{2s-1}\delta^2 > \frac{2^{s+1}\delta}{\sqrt{2km}} \max\{24c_1, 29\}$$

by (8), therefore, we can apply Corollary 3.5 to each \mathbb{P}_s . This gives

$$\begin{aligned} \sum_{s=0}^{\infty} \mathbb{P}_s & \leq \sum_{s=0}^{\infty} 4 \exp\left\{-\frac{mk2^{4s-2}\delta^4}{c_3 2^{2s+2}\delta^2}\right\} \\ & = \sum_{s=0}^{\infty} 4 \exp\left\{-\frac{mk2^{2s-4}\delta^2}{c_3}\right\} \\ & = 4 \exp\left\{-\frac{mk\delta^2}{16c_3}\right\} + 4 \exp\left\{-\frac{mk\delta^2}{4c_3}\right\} + \sum_{s=0}^{\infty} 4 \exp\left\{-\frac{2mk\delta^2 2^s}{c_3}\right\} \\ & = 4 \exp\left\{-\frac{mk\delta^2}{16c_3}\right\} + 4 \exp\left\{-\frac{mk\delta^2}{4c_3}\right\} + 4 \left(1 - \exp\left\{-\frac{2mk\delta^2}{c_3}\right\}\right)^{-1} \exp\left\{-\frac{2mk\delta^2}{c_3}\right\}. \end{aligned}$$

Then, by condition (8), we have that

$$4 \left(1 - \exp\left\{-\frac{2mk\delta^2}{c_3}\right\}\right)^{-1} < 5$$

and the first exponential is the largest so we have the result. \square

Finally, we can use the Lipschitz behavior of the function Φ_h , combined with the bound $h^* < M$ to derive our main result.

Proof of Theorem 1.3. The function $\Phi_h(n)$ is well behaved. In particular, we have that for some $c(n, M)$,

$$c(n, M)|h - h'| \leq |\Phi_h(n) - \Phi_{h'}(n)|$$

for all $h, h' < M$ and every n . This is easily verified, though it is necessary to calculate $c(n, M)$ numerically. Therefore,

$$\frac{|h - h'|}{\sqrt{k}} \sqrt{\sum_{\ell=1}^k c^2(n_{\ell}, M)} \leq \frac{1}{\sqrt{k}} \sqrt{\sum_{\ell=1}^k (\Phi_h(n_{\ell}) - \Phi_{h'}(n_{\ell}))^2}.$$

So setting $c_2 = \frac{1}{k} \sum_{\ell=1}^k c^2(n_{\ell}, M)$ and applying Theorem 3.6 gives the result. \square

Proof of Theorem 1.4. Define $A = \{\sup_{f \in \mathcal{F}} |R_n(f) - \hat{R}_n(f)| > \rho\}$ and $B = \{h^* < \hat{h} + \delta\}$, then we are interested in controlling $\mathbb{P}(A)$. By the law of total probability, we have³

$$\begin{aligned}\mathbb{P}(A) &= \mathbb{P}(A \mid B)\mathbb{P}(B) + \mathbb{P}(A \mid B^c)\mathbb{P}(B^c) \\ &\leq \mathbb{P}(A \mid B)\mathbb{P}(B) + \mathbb{P}(B^c) \\ &= 4GF(\hat{h} + \delta, 2n) \exp\{-n\rho^2\}(1 - \varphi) + \varphi\end{aligned}$$

□

4 Discussion

In this paper, we showed how to derive generalization error bounds from the estimated rather than actual VC dimension of a function class \mathcal{F} . Our method uses the simulation procedure proposed by Vapnik et al. [10] for the estimates. Empirical process theory for nonparametric least squares regression shows that these estimates \hat{h} concentrate around the truth h^* with high probability. The resulting bounds can be used for model selection as well as to characterize the finite-sample predictive ability of the model \hat{f} chosen through empirical risk minimization.

The algorithm outlined here is not the only way to estimate VC dimension. Shao et al. [4] modify Algorithm 1 in light of ideas from experimental design, varying the number of replications m with the design point n_ℓ , and show that this improves the estimates of the VC dimension. Modifying our empirical process techniques to use this improved estimator would be desirable, but the extension is nontrivial.

As mentioned in the introduction, there are many other methods for measuring the richness of a model class. Rademacher complexity in expectation is difficult or impossible to calculate, but it has an obvious empirical counterpart for which concentration results already exist thereby allowing for tight data-based generalization error bounds. However, Rademacher complexity cannot be used with unbounded loss functions. VC dimension, while discussed here in the context of classification, generalizes to regression problems with unbounded loss as long as appropriate moment conditions are satisfied. Hence, our technique will apply in these settings as well. Indeed, since VC dimension is a property of the class of prediction functions and not the data-generating process, and finite VC dimension has recently [1] been shown to characterize learning from ergodic sources, it may be possible to use our procedure as part of an algorithm for bounding prediction risk on dependent data.

References

- [1] ADAMS, T. M., AND NOBEL, A. B. (2010), “Uniform convergence of Vapnik-Chervonenkis classes under ergodic sampling,” *Annals of Probability*, **38**, 1345–1367.
- [2] BARTLETT, P. L., AND MENDELSON, S. (2002), “Rademacher and Gaussian complexities: Risk bounds and structural results,” *Journal of Machine Learning Research*, **3**, 463–482.
- [3] POLLARD, D. (1984), *Convergence of Stochastic Processes*, Springer Verlag, New York.

³Technically, the data come from the distribution μ while Algorithm 1 uses some other distribution, say ν , to generate simulated data. Therefore, the probability statement in this theorem is with respect to the product measure $\mu \times \nu$. For the result to hold, we must have that μ and ν are measures over the same probability space $\mathcal{Y} \times \mathcal{X}$ and that the real and simulated data are statistically independent.

- [4] SHAO, X., CHERKASSKY, V., AND LI, W. (2000), “Measuring the VC-dimension using optimized experimental design,” *Neural computation*, **12**(8), 1969–1986.
- [5] VAN DE GEER, S. (1990), “Estimating a regression function,” *Annals of Statistics*, **18**(2), 907–924.
- [6] VAN DE GEER, S. (2000), *Empirical Processes in M-estimation*, Cambridge University Press, Cambridge, UK.
- [7] VAPNIK, V. N. (1998), *Statistical learning theory*, John Wiley and Sons, New York.
- [8] VAPNIK, V. N. (2000), *The Nature of Statistical Learning Theory*, Springer Verlag, New York, 2nd edn.
- [9] VAPNIK, V. N., AND CHERVONENKIS, A. J. (1991), “The necessary and sufficient conditions for consistency of the method of empirical risk,” *Pattern Recognition and Image Analysis*, **1**(3), 284–305.
- [10] VAPNIK, V. N., LEVIN, E., AND LECUN, Y. (1994), “Measuring the VC-dimension of a learning machine,” *Neural Computation*, **6**(5), 851–876.